Icahn
School of
Medicine at
Mount
Sinai

Introduction to coding with R

Part I

Joselyn Chavez

04/12/2022

Data structures in R

- Vectors
- Matrices
- Data frames
- Lists
- Functions

Vectors

Creating a vector

Using the assignment operator

For one value

```
my_vector <- 10
my_vector <- "a"
```

Using the combine function

For two or more values

```
my_vector <- c(1,10,25,30)
my_vector
```

```
## [1]  1 10 25 30
```

```
my_vector <- c("a","b","c")
my_vector
```

```
## [1] "a" "b" "c"
```

Using the seq() function

```
my_vector <- seq(1:10)
my_vector
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
my_vector <- seq(from = 0, to = 10, by = 2)
my_vector
```

```
## [1]  0  2  4  6  8 10
```

Vector features

- Vectors have only one dimension (length)

```
my_vector <- c(1,2,3,4)
length(my_vector)
```

```
## [1] 4
```

- All vector components must be the same type
    - Numeric
    - Integer
    - Double
    - Character
    - Factor
    - Logical

- Numeric

```
x_num <- c(1, 2, 3)
class(x_num)
```

```
## [1] "numeric"
```

- Integer

```
x_int <- c(1L, 2L, 3L)
class(x_int)
```

```
## [1] "integer"
```

- Double

```
x_dbl <- c(1, 2.5, 3.1)
typeof(x_dbl)
```

```
## [1] "double"
```

- Character

```
x_chr <- c("a", "b", "c")
class(x_chr)
```

```
## [1] "character"
```

- Factor

```
x_fct <- factor("a","b","c")
class(x_fct)
```

```
## [1] "factor"
```

- Logical

```
x_log <- c(TRUE, FALSE, TRUE)
class(x_log)
```

```
## [1] "logical"
```

- R finds a way to unify data type when there is more than one per vector

```
x <- c(1, "a", TRUE)
x
```

```
## [1] "1"    "a"    "TRUE"
```

```
class(x)
```

```
## [1] "character"
```

Missing values

- NA

```
x <- c(1, "a", TRUE, NA)
x
```

```
## [1] "1"    "a"    "TRUE" NA
```

- NaN

```
x <- c(10, -1, NA)
log(x)
```

```
## [1] 2.302585      NaN       NA
```

How to access vector elements?

Using integer as index

Vector index in R starts from 1

```
x <- c(1,2,3,4,5)
x
```

```
## [1] 1 2 3 4 5
```

```
x[3] # Extract the third element
```

```
## [1] 3
```

```r
x <- c(1,2,3,4,5)

# Extract index from 3 to 5
x[3:5]
```

```
## [1] 3 4 5
```

```r
x <- c("a","b","c","d","e")

# Extract index 2 and 5
x[c(2,5)]
```

```
## [1] "b" "e"
```

Using the name as index

```
x <- c(1,3,10)
names(x)
```

```
## NULL
```

```
x <- c("first"= 1,"second"=3,"third"=10)
x
```

```
##  first second  third
##      1      3     10
```

```
names(x)
```

```
## [1] "first"  "second" "third"
```

```r
x <- c(1,3,10)

names(x) <- c("first", "second", "third")
x
```

```
##  first second  third
##      1      3     10
```

```r
x["second"]
```

```
## second
##      3
```

```r
x[c("first","third")]
```

```
## first third
##     1    10
```

Using logical evaluation as index

```
x <- seq(1:10)
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
x < 5
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALS
```

```
x[x < 5]
```

```
## [1] 1 2 3 4
```

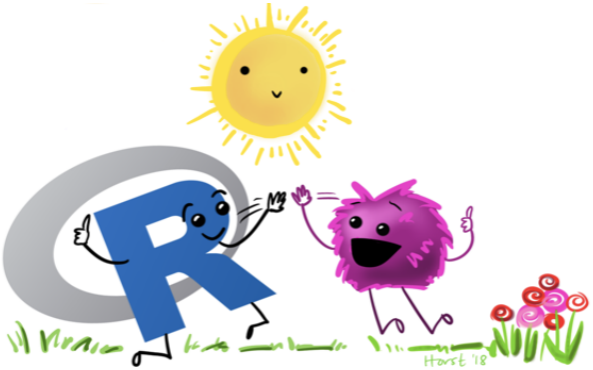
```r
x <- c("a","a","b","c","c","c")

x == "c"
```

```
## [1] FALSE FALSE FALSE  TRUE  TRUE  TRUE
```

```r
x[x == "c"]
```

```
## [1] "c" "c" "c"
```

Thanks!



Ilustration by Allison Horst